[AIR-DSP Instruction Pipeline Architecture v1.0]

Purpose:

Define a hardware-level command pipeline architecture for audio DSP

Based on modular instruction sets mapped across RISC-V (integer DSP) and SPIR-V (GPU shaders)

Achieve sub-1ms end-to-end latency via parallelized, format-adaptive execution

Note: Designed by 제현 via GPT-4o prompt-based vibe coding. This document outlines the execution architecture, not just a software structure.

[Pipeline Overview - SoC Command Flow]

Stage 0: PCM Fetch

Input: 16-bit PCM at 48kHz

Operation: Buffer into register bank

Control: sample_count, frame_sync

Stage 1: Preprocessing

Instruction: AIR_AGC, AIR_NS

Format: FP16 or FP32

Backend: RISC-V (scalar or SIMD)

Stage 2: MDCT + Windowing

Instruction: AIR_MDCT16, AIR_MULQ31

Input: Preprocessed frame

Format: Q31 or BFP16

Backend: RISC-V

Stage 3: Quantization (SPIR Block)

Shader: spir_quantize_subband, spir_vecmap_q31

Operation: Normalize → Scale → Round

Format: Q31 → int8/int16

Backend: SPIR-V (Compute Shader)

Stage 4: Packet Encoding

Instruction: AIR_QUANT8, AIR_PACK

Input: Quantized bins

Format: int8 + config header

Backend: RISC-V DSP

Stage 5: Checksum + Output

Instruction: AIR_CRC

Output: LC3-style payload packet

Target: BLE, DAC, or file buffer

[Execution Pipeline Spec]

IF: Fetch AIR Instruction

ID: Decode opcode + format flags

EX: Dispatch to RISC-V or SPIR-V unit

MEM: Access buffer/registers

WB: Write to output, memory or I/O

Supports:

Format switching at ID stage (FP↔Q31↔BFP)

Latency priority control per stage (via LAT_CTL reg)

Integration with runtime fallback (SW/GPU hybrid)

[Core AIR Instruction Examples] AIR_AGC rd, rs   ; automatic gain control (FP) AIR_MDCT16 rd, rs   ; 16-point MDCT transform AIR_MULQ31 rd, rs1, rs2 ; Q31 multiply AIR_VECMAP rd, rs   ; FP → Q31 normalizer AIR_QUANT8 rd, rs   ; quantize to 8-bit integer AIR_PACK rd, rs1, rs2 ; packet header + payload join AIR_CRC rd, rs   ; 32-bit checksum

---

[Pipeline Highlights]

Hardware-level instruction scheduling

Modular command units (FP, Q31, BFP aware)

Dual-backend optimized (SPIR-V + RISC-V)

Ready for implementation in FPGA softcore DSP or simulation

Author: 제현 Date: 2025-10-15 Version: AIR-DSP v1.0